



PXE Engineering

Intel Architecture Labs

PXE for Linux Software Development Kit Instructions

Version 3.0

May 4, 1999

Intel Corporation assumes no responsibility for errors or omissions in this guide. Nor does Intel make any commitment to update the information contained herein.

- * Other product and corporate names may be trademarks of other companies and are used only for explanation and to the owners' benefit, without intent to infringe.

Copyright © 1998, 1999 Intel Corporation. All rights reserved.
Intel Corporation, 5200 N.E. Elam Young Parkway, Hillsboro, OR 97124-6497

Contents

| | |
|---|-----------|
| 1. SDK LICENSE | 1 |
| 2. SDK INSTALLATION AND BUILD INSTRUCTIONS | 4 |
| 2.1 EQUIPMENT NEEDED..... | 4 |
| 2.2 PREPARATION..... | 4 |
| 2.3 INSTALL REDHAT 5.2 LINUX..... | 1 |
| 2.4 BOOTP/NFS SERVER CONFIGURATION (PART 1) | 2 |
| 2.5 PXE SERVER SETUP..... | 7 |
| 2.6 BOOTP/NFS SERVER CONFIGURATION (PART 2) | 11 |
| 3. CONFIGURATION OF PXE AND MTFTP DAEMONS | 13 |
| 3.1 CONFIGURATION FILE FOR PXE DAEMON..... | 13 |
| 3.2 CONFIGURATION FILE FOR MTFTP DAEMON | 13 |
| 4. REVISION HISTORY | 14 |

1. SDK License

INTEL PREBOOT EXECUTION ENVIRONMENT (PXE) SOFTWARE DEVELOPMENT KIT (SDK) COPYRIGHT LICENSE AND DISTRIBUTION AGREEMENT

IMPORTANT: READ BEFORE COPYING, INSTALLING OR USING.

BY USING THIS SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS AGREEMENT. DO NOT USE THIS SOFTWARE UNTIL YOU HAVE CAREFULLY READ AND AGREED TO THE FOLLOWING TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE SOFTWARE FILES AND ANY ACCOMPANYING ITEMS AND DO NOT COPY, INSTALL OR USE THE SOFTWARE. .

IF YOU USE THIS SOFTWARE, YOU WILL BE BOUND BY THE TERMS OF THIS AGREEMENT AS FOLLOWS:

1. Definitions.

Affiliated Corporations mean the parent corporation and any subsidiary corporation which is wholly owned or majority owned by the parent corporation.

Binary Code means software in binary code form as provided to Licensee by Intel and software in binary code form compiled by Licensee from Support Code and permitted Licensee modifications to Support Code.

Documentation means all documentation provided by Intel under this Agreement.

Licensee Product means only products that incorporate all or a portion of the Materials, and that are manufactured by or for, and distributed by Licensee subject to the licenses granted by Intel under this Agreement.

Licensee means the individual or entity identified as the Licensee upon ordering these Materials from Intel. If Licensee is a United States Corporation, then Licensee shall include Affiliated Corporations, provided that such Affiliated Corporations agree to the terms and conditions of this Agreement and provided that Licensee remains solely responsible for the duties and obligations of Licensee under this Agreement.

Materials include Binary Code, Support Code and Documentation as provided by Intel to Licensee under this Agreement.

Specifications means the "Wired for Management Baseline Specification Version 2.0", and the "Preboot Execution Environment Specification Version 2.0" and subsequent updates to these specifications.

Support Code means all source code included in this package, all of which is provided to Licensee by Intel subject to the licenses granted hereunder.

2. **Copyright License.** Subject to the terms and conditions of this Agreement, Intel Corporation ("Intel") grants to Licensee a non-exclusive, non-transferable, royalty-free license, under Intel copyrights, to the Materials as set forth in this Agreement. Grant of the license is subject to Licensee's strict compliance with the following conditions:

A. Licensee may:

(i) Support Code: Reproduce and modify the Support Code, for internal use only, solely for the purpose of (a) integrating Binary Code and Documentation with Licensee Product, (b) creating bug fixes and error corrections to Support Code, and (c) compiling Support Code and permitted modifications thereto into Binary Code for internal use only, except as provided in Section 2.A(ii).

(ii) Binary Code: Reproduce and distribute Binary Code in binary form only, including binary code for bug fixes, error corrections and permitted modifications that are made to Support Code pursuant to Section 2.A(i) above, only incorporated into Licensee Product.

(iii) Documentation: Reproduce and modify the Documentation for internal use only. Notwithstanding the foregoing, Licensee may distribute Documentation for Binary Code, and modifications to such Documentation, only to Licensee's customers in conjunction with Licensee Product.

B. Licensee may not:

(i) Distribute the Materials except as provided in Section 2.A.

(ii) Permit others to disassemble, decompile, or otherwise reverse engineer the Binary Code, in whole or in part.

(iii) Use, copy, modify, rent, sell or transfer the Materials or any portion thereof except as provided in Section 2.A.

(iv) Make any statements that the Licensee Product or the Materials are certified by, or that performance is guaranteed or verified in any regard, by Intel.

(v) Use Intel's name or trademarks to market Licensee Products or any other product without Intel's prior written consent.

3. **Ownership.** Licensee acknowledges that title to the Materials remain the property of Intel and its vendors. Subject to Intel's ownership of the Materials, however, Licensee Products shall be the property of Licensee.
4. **Copyright Protection.** The Materials are copyrighted and are protected by United States copyright laws and international treaty provisions. Licensee agrees to prevent any unauthorized copying of the Materials. Licensee shall not remove or obscure any of Intel's or its vendors' copyright notices or other proprietary notices from the Materials.
5. **Copyright Attribution.** Each Licensee Product shall display, in a user-visible location such as a start-up screen or an "About" box, and in a message to be displayed on-screen at the time Licensee Products are installed on an end-user system, the following legend, which shall be displayed for at least two (2) seconds

"PXE Software Copyright © 1999 Intel Corporation"

6. **Compliance with Specifications.** Licensee shall use commercially reasonable efforts to ensure that Licensee Products comply with the Wired For Management Baseline Specifications ("Specifications") and any updates thereto and any reasonably related test suites (implemented by Intel from time to time at Intel's sole discretion) within a reasonable time after such updates to the Specifications become available publicly.
7. **Support.** Intel has no obligation to provide support of any kind to Licensee regarding use of the Materials. Intel may, at its sole discretion, create upgrades and updates to correct errors and bugs. Intel may make any such upgrades and updates available to Licensee on electronic bulletin board service or on the Internet and Intel shall have no obligation to notify Licensee of the availability of any such upgrade or update. Such upgrades and updates shall be considered part of the Materials, and shall be subject to this Agreement. Licensee shall be solely responsible to its customers for any update or support obligation or other liability that may arise from Licensee's distribution of the Materials or modified Materials.
8. **Confidentiality.** Support Code may include trade secrets of Intel. Except as permitted by this Agreement, Licensee shall not disclose or otherwise make any part of Support Code (whether or not modified by Licensee) available, in any form, to any person other than Licensee's employees whose job performance requires such access. Licensee shall instruct all such employees on these obligations with respect to use, copying, protection, and confidentiality of Support Code. Even after this Agreement terminates, the obligations of this section shall remain in effect until the Support Code rightfully becomes publicly known.

9. **Reservation of Rights.** Except as expressly provided herein, Intel does not grant any express or implied right or license to Licensee under any Intel patents, copyrights, trademarks, or trade secret information.

10. **Limited Warranty.** The Materials are provided "AS IS" without warranty of any kind.

INTEL MAKES NO WARRANTIES OF ANY KIND, WHETHER EXPRESS, IMPLIED OR STATUTORY, AND EXPRESSLY DISCLAIMS WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, AND FITNESS FOR ANY PARTICULAR PURPOSE. INTEL WILL NOT PROVIDE ANY SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES. INTEL WILL NOT PROVIDE ANY UPDATES, ENHANCEMENTS OR EXTENSIONS.

Intel further does not warrant or assume any responsibility for the accuracy or completeness of the information, text, graphics or other items contained in the materials. Intel may make changes to the Materials or to any products described therein at any time without notice.

11. **Limitation of Liability.**

INTEL, ITS VENDORS AND SUPPLIERS SHALL HAVE NO LIABILITY OF ANY KIND, INCLUDING BUT NOT LIMITED TO LIABILITY FOR ANY INDIRECT, SPECULATIVE, CONSEQUENTIAL, INCIDENTAL, AND SPECIAL DAMAGES, INCLUDING WITHOUT LIMITATION, PROCUREMENT COSTS, LOSS OF USE, BUSINESS INTERRUPTIONS, AND LOSS OF PROFITS, IRRESPECTIVE OF WHETHER LICENSEE HAS ADVANCE NOTICE OF THE POSSIBILITY OF ANY SUCH DAMAGES.

12. **Indemnity.** Licensee shall indemnify, defend and hold Intel harmless from and against any claims or lawsuits, including attorney fees that arise or result from the use or distribution of Licensee Products.

13. **Termination.** Intel may terminate this Agreement at any time if Licensee is in material breach of any of its terms and conditions. Upon termination, Licensee shall immediately destroy the Materials or return all copies of the Materials to Intel along with any copies thereof.

14. **U.S. Government Restricted Rights.** The Materials are provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the government is subject to restrictions as set forth in FAR 52.227-14 and DFARS 252.227-7013 et. seq. or its successor. The use of this product by the Government constitutes acknowledgment of Intel's proprietary rights in the product.

15. **Applicable Law and Notice.** Any claim arising under or relating to this Agreement shall be governed by the internal substantive laws of the State of Delaware or federal courts located in Delaware, without regard to principles of conflict of laws. Each party hereby agrees to jurisdiction and venue in the courts located in the State of Delaware for all disputes and litigation arising under or relating to this Agreement. This provision is meant to comply with 6 Del. C. Section 2708(a).

16. **Severability.** The terms and conditions stated in this Agreement are declared to be severable. If any paragraph, provision, or clause in this Agreement shall be found or be held to be invalid or unenforceable in any jurisdiction in which this Agreement is being performed, the remainder of this Agreement shall be valid and enforceable and the parties shall use good faith to negotiate a substitute, valid, and enforceable provision which most nearly effects the parties' intent in entering into this Agreement.

2. SDK Installation and Build Instructions

2.1 Equipment Needed

2.1.1 Hardware you need to build and test PXE

- BOOTP/NFS Server: 16+ MB DRAM, 2+ GB Hard drive (HD), Network Interface Card (NIC)
- PXE Server(s): 16+ MB DRAM, 2+ GB HD, NIC
- Client(s): 16+ MB DRAM, NIC
- Hubs / Routers (as needed)
- Network Analyzer (recommended)

2.1.2 Software you need to build and test PXE

- RedHat Linux 5.2 CD#1
- Diskette containing PXE Linux SDK archive (pxesdk.tgz)
- Diskette containing patch-non640k-2.0.36.gz patch file (see next section)

2.2 Preparation

2.2.1 Download patch-non640k-2.0.36

This patch should be located at (or near) here:

<http://metalab.unc.edu/pub/Linux/system/boot/patch-non640k-2.0.36.README>

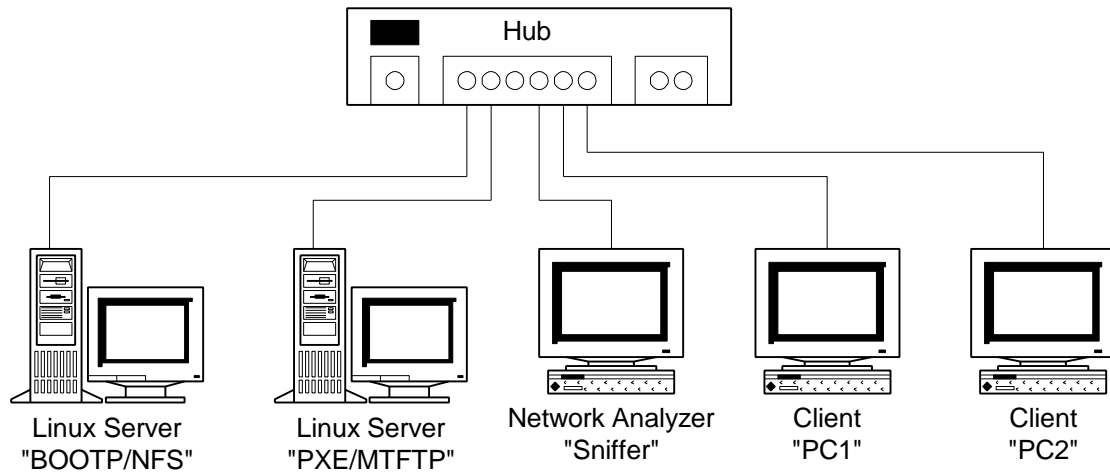
<http://metalab.unc.edu/pub/Linux/system/boot/patch-non640k-2.0.36.gz>

<http://metalab.unc.edu/pub/Linux/system/boot/patch-non640k-2.0.36.lsm>

This patch is needed when remote booting a kernel that uses the PXE Universal Network Driver Interface (UNDI). It will be used later in this document. Put these files on your "patch" diskette.

2.2.2 Setup PXE SDK test network

This suggested network setup is a good starting point when installing and configuring the PXE SDK for the first time. Keep it simple. Do not use routers and do not connect to any other network segments.



It is a good idea to write down some information about the machines on your network at this time. The form on the next page can be used to record this information. Keep this information handy, you will use it later when installing and configuring the Linux servers.

| <u>Field Names</u> | <u>Your Setup</u> | <u>Examples</u> |
|-------------------------|-------------------------------|----------------------|
| Network | | |
| Subnet Mask: | ____.____.____.____ | 255.255.255.0 |
| Domain: | _____ | pxetest.net |
| BOOTP/NFS Server | | |
| Name: | _____ | bootpnfs.pxetest.net |
| Alias: | _____ | bootpnfs |
| IP Address: | ____.____.____.____ | 192.215.102.240 |
| PXE/MTFTP Server | | |
| Name: | _____ | pxemtftp.pxetest.net |
| Alias: | _____ | pxemtftp |
| IP Address: | ____.____.____.____ | 192.215.102.241 |
| Sniffer | | |
| Name: | _____ | snort.pxetest.net |
| Alias: | _____ | snort |
| IP Address: | ____.____.____.____ | 192.215.102.242 |
| Client: PC1 | | |
| Name: | _____ | pc1.pxetest.net |
| Alias: | _____ | pc1 |
| IP Address: | ____.____.____.____ | 192.215.102.10 |
| MAC Address: | ____:____:____:____:____:____ | 00:11:22:33:44:55 |
| Client: PC2 | | |
| Name: | _____ | pc2.pxetest.net |
| Alias: | _____ | pc2 |
| IP Address: | ____.____.____.____ | 192.215.102.11 |
| MAC Address: | ____:____:____:____:____:____ | ff:ee:dd:cc:bb:aa |

2.3 Install RedHat 5.2 Linux

This section must be completed twice, once for the BOOTP/NFS Server and once for the PXE Server. These instructions assume that you know how to install RedHat Linux and do not need to be walked through the process in minute detail (besides, some of the minute details change depending on your hardware configuration).

When a window name matches a section header below, make sure the items that are marked in **bold** are selected.

2.3.1 Installation Class Window

Select:

Custom

2.3.2 Components to Install Window

The PXE SDK was tested with the following components selected. Most of these were already on by default when we installed our RedHat Linux, maybe not all are needed, but we know the PXE SDK works with these components selected.

X Window System
DOS/Windows Connectivity
File Managers
Networked Workstations
NFS Server
SMB (Samba) Connectivity
Network Management Workstation
C Development
Development Libraries
C++ Development
Extra Documentation

2.3.3 Boot Protocol Window

Select:

Static IP address

2.3.4 Configure TCP/IP Window

Enter values for IP address and Netmask. Default gateway (IP) may be left blank if you are not booting across routers. Primary nameserver may be left blank if you are not using DNS.

2.3.5 Services Window

The PXE SDK was tested with the following services selected. Most of these were already on by default when we installed our RedHat Linux, maybe not all are needed, but we know the PXE SDK works with these services selected.

| | | |
|-----------------|------------------|----------------|
| atd | keytabled | pcmcia |
| crond | linuxconf | portmap |
| gpm | network | random |
| inet | nfs | smb |
| kernelld | nfsfs | syslog |

2.3.6 Finish The Installation

When you are done with the installation the system will restart. Install RedHat Linux on the next server.

2.4 BOOTP/NFS Server Configuration (part 1)

The BOOTP/NFS Server configuration has been split into two parts to make it easier to locate configuration errors, if any are made. Please double check your changes, reset machines when requested and run the test at the end of the section.

2.4.1 Install PXE Linux SDK

There are a few sample files included in the PXE Linux SDK archive ([pxesdk.tgz](#)) that will help you to configure your BOOTP and NFS daemons. It is recommended that you install the PXE Linux SDK on this machine, but do not build and install the PXE or MTFTPD daemons (this will be done on the [PXE Server](#)).

```
mkdir /pxe
cd /pxe
mount /dev/fd0 /mnt/floppy
tar xfovz /mnt/floppy/pxesdk.tgz
umount /mnt/floppy
```

2.4.1.1 PXE Linux SDK Archive Contents

Note: The organization of these files is in flux. Expect changes in the next release.

Additional note: The files in the [./server/nbp/](#) directory must be built on a machine running [Microsoft 16-bit C/C++ v1.52c](#). These files are used to create the [./server/linux.0](#) bootstrap program.

| Filename | Description |
|--|--|
| ./makefile | Used to build & install PXE and MTFTP daemons and libraries. |
| ./samples/etc/bootptab | Sample bootptab to be placed on BOOTP server. |
| ./samples/etc/exports | Sample exports to be placed on NFS server. |
| ./samples/etc/fstab | Sample fstab to be placed in remote client tree. |
| ./samples/etc/hosts | Sample hosts to be place on all servers and remote client trees. |
| ./samples/etc/hosts.allow | Sample hosts.allow to be placed on NFS server. |
| ./samples/etc/rc.d/rc.sysinit | Sample rc.sysinit to be placed in remote client tree. |
| ./samples/usr/src/linux/make.config | Minimal ".config" file for building remote boot kernel image. |
| ./client/patches/patch-pxeundi-2.0.36.gz | Patch to include UNDI driver into kernel source tree. |
| ./server/linux.0 | PXE initial environment layer for Linux remote boot. This is the first file downloaded to the client. This file will download linux.1. Linux.1 is the kernel image that you will create later. |
| ./server/makefile | Used to build, install and remove PXE & MTFTP daemons. |
| ./server/classes/apstring.cc | String class. |
| ./server/classes/cdhcp.cc | DHCP packet class. |
| ./server/classes/cipaddr.cc | IP address class. |
| ./server/classes/cproxreg.cc | Proxy registry class. |
| ./server/classes/cpxedebg.cc | PXE debug message class. |
| ./server/classes/makefile | Used to build, install & remove class library. |
| ./server/classes/registry.cc | Registry class. |
| ./server/include/apstring.h | String class definition. |
| ./server/include/bootp.h | BOOTP packet definition. |
| ./server/include/cdhcp.h | DHCP class definition. |
| ./server/include/cipaddr.h | IP address class definition. |
| ./server/include/cproxreg.h | Proxy registry class definition. |
| ./server/include/cpxedebg.h | PXE debug class definition. |
| ./server/include/dhcp.h | DHCP option definitions. |
| ./server/include/dhcppkt.h | DHCP packet definition. |
| ./server/include/optsdll.h | Option parser class library definition. |
| ./server/include/parsedll.h | Parse class library definition. |
| ./server/include/pktdata.h | Packet data class library definition. |

| Filename | Description |
|--|---|
| ./server/include/pxe.h | DHCP PXE vendor option definitions. |
| ./server/include/pxemisc.h | PXE type definitions. |
| ./server/include/registry.h | Registry class definition. |
| ./server/mtftpd/makefile | Used to build, install and remove MTFTP daemon. |
| ./server/mtftpd/mtftpd.c | MTFTP daemon source. |
| ./server/mtftpd/mtftpd.conf | MTFTP daemon configuration file. |
| ./server/mtftpd/mtftpdconf.c | MTFTP daemon configuration file reader source. |
| ./server/mtftpd/tftpsubs.c | TFTP subroutines. |
| ./server/mtftpd/tftpsubs.h | TFTP subroutine definitions. |
| ./server/services/makefile | Used to build, install & remove PXE daemon. |
| ./server/services/pxe.conf | PXE daemon configuration file. |
| ./server/services/pxeservice.cc | PXE service source. |
| ./server/services/apitestopts/apitestopts.cc | API test option parser class source. |
| ./server/services/apitestopts/makefile | API test option parser class makefile. |
| ./server/services/bstrapopts/bstrapopts.cc | Bootstrap option parser class library source. |
| ./server/services/bstrapopts/makefile | Bootstrap option parser class library source. |
| ./server/services/pxeclienttester/makefile | PXE client tester class library makefile. |
| ./server/services/pxeclienttester/pxeclienttester.cc | PXE client tester class library source. |
| ./server/services/pxeparser/makefile | PXE packet parser class library makefile. |
| ./server/services/pxeparser/pxeparser.cc | PXE packet parser class library source. |
| ./server/nbp/_nul_.asm | Bootstrap program entry point and NULL pointer. |
| ./server/nbp/apm.asm | APM BIOS routines. |
| ./server/nbp/arp.h | ARP packet header. |
| ./server/nbp/bis.h | BIS structures & constants. |
| ./server/nbp/bootp.h | BOOTP packet header. |
| ./server/nbp/build.bat | Batch used to setup 16-bit environment under WinNT DOS box. |
| ./server/nbp/cdefs.h | PXE ROM prototypes & constants. |
| ./server/nbp/dhcp.h | DHCP structures & tag constants. |
| ./server/nbp/dhcputil.h | DHCP library routine definitions. |
| ./server/nbp/dmi.h | DMI BIOS structures & constants. |
| ./server/nbp/ether.h | Ethernet packet header. |
| ./server/nbp/getpkt.c | Get cached packets from PXE APIs. |
| ./server/nbp/igmp.h | IGMP packet header. |
| ./server/nbp/iobuf.h | Complete packet stack. |
| ./server/nbp/ip.h | IP packet header. |
| ./server/nbp/lisa_heap.h | Simple heap prototypes. |
| ./server/nbp/lisa_io.h | Simple console prototypes. |
| ./server/nbp/lisa_jmp.h | Setjmp/Longjmp prototypes. |
| ./server/nbp/lisa_mem.h | Memory copy/move prototypes. |
| ./server/nbp/lisa_misc.h | Misc. services. |
| ./server/nbp/lisa_str.h | String copy/move prototypes. |
| ./server/nbp/makefile | Build linux.0. |
| ./server/nbp/nbp.h | PXE common NBP header. |
| ./server/nbp/nbpentry.c | Linux.0 bootstrap program C entry point & main code. |
| ./server/nbp/pxe.c | PXE API initialization and call routines. |
| ./server/nbp/pxe.h | PXE structures and tag constants. |
| ./server/nbp/pxe_api.h | PXE API structures and constants. |
| ./server/nbp/pxe_cmh.h | PXE status and error codes. |
| ./server/nbp/pxeid.h | PXE ROM ID structures. |
| ./server/nbp/pxever.h | PXE ROM version numbers. |
| ./server/nbp/redhat.mak | Build linux.0. |
| ./server/nbp/setup.asm | Linux kernel setup routine. |
| ./server/nbp/sysid.h | SYSID/UUID structures. |
| ./server/nbp/tftp.h | TFTP packet header. |
| ./server/nbp/tftp_api.h | TFTP API structures and constants. |
| ./server/nbp/tune.h | Timeout constants. |

| Filename | Description |
|--------------------------------------|------------------------------------|
| <code>./server/nbp/udp.h</code> | UDP packet header. |
| <code>./server/nbp/udp_api.h</code> | UDP API structures and constants. |
| <code>./server/nbp/undi_api.h</code> | UNDI API structures and constants. |

2.4.2 Install BOOTP Package

Login to your server as root.

The BOOTP daemon package is not included in the default RedHat Linux installation. You will need to mount your RedHat CD#1 and install the BOOTP daemon package.

```
mount /dev/cdrom /mnt/cdrom
rpm -Uvh /mnt/cdrom/RedHat/RPMS/bootp*
umount /mnt/cdrom
```

(Ignore the error message about bootpc already being installed.)

2.4.3 Create /etc/bootptab

If you do not already have an /etc/bootptab file, you can start with the sample file /pxe/samples/etc/bootptab.

```
cp /pxe/samples/etc/bootptab /etc/bootptab
```

You will need the MAC address(es) of your client machine(s). (The MAC address is displayed on the screen by the PXE boot ROM when the machine is turned on.) When you have the MAC address(es), you will need to edit the /etc/bootptab file and change some of the information. A sample /etc/bootptab file (with two client machines) is shown here.

```
pc1:ht=1:ha=112233445566:ip=1.2.3.10:rp=/nfsroot/pc1:sm=255.255.255.0
pc2:ht=1:ha=0xffeeddccbbaa:ip=1.2.3.11:rp=/nfsroot/pc2:sm=255.255.255.0
```

If you want to know more about the fields in the /etc/bootptab file type “man bootptab” at the command line. For now, we just want you to enter some information about your network, NFS server and client machines.

You may change the machine names to suit you. We use “pc1” and “pc2”. If you change a machine name, you should change the NFS root path (“rp=” field) to match the new name.

Do not change the network hardware type (“ht=” field).

Each client machine must have a unique MAC address (“ha=” field). This MAC address must be the same address that is displayed on the client screen by the PXE boot ROM.

Each client machine must have a unique IP address (“ip=” field). Pick an address that is not used by any other machine on your network.

The subnet mask (“sm=” field) should be the same for all clients on the same subnet.

2.4.4 Edit /etc/inetd.conf

Edit the /etc/inetd.conf file and locate the line below (search for “#bootps”):

```
#bootps          dgram udp    wait  root    /usr/sbin/tcpd    bootpd
```

Uncomment the line (remove the tic-tac-toe) so it looks like this:

```
bootps          dgram udp    wait  root    /usr/sbin/tcpd    bootpd
```

The BOOTP daemon will now start when the machine is reset.

2.4.5 Edit /etc/rc.d/rc.local

We need to get the IP broadcast packets routed to the BOOTP daemon. This is done by adding the following line to the end of the /etc/rc.d/rc.local file:

```
route add -host 255.255.255.255 eth0
```

2.4.6 Reset The Server

Wait for the server to restart and login as root.

2.4.7 Boot A Client

Turn on one of the client machines that you created an entry for in the `/etc/bootptab` file. Since we did not setup the `/etc/bootptab` to supply a boot filename *and* there is no PXE server running on the network, you will get the following error (it takes about one minute for the PXE boot ROM to timeout and give you the error):

```
PXE-E53: No boot filename received
```

This means that the client received a BOOTP reply, but did not get a boot filename. If you get this error, you have correctly initialized your BOOTP daemon. The other two errors you might see are:

```
PXE-E51: No DHCP or BOOTP offers received
```

```
PXE-E61: Media test failure, check cable.
```

PXE-E61 means that the client machine is not plugged into the network, or there is some kind of hardware problem (usually a bad cable). PXE-E51 means that the client did not get a BOOTP reply. This probably means that your BOOTP was not installed/setup correctly (without a network analyzer this is difficult to determine).

The next section discusses BOOTP troubleshooting, it can be skipped if you got the PXE-E53 error.

2.4.8 BOOTP Troubleshooting

Start your network analyzer and reboot the client machine. You should see five DHCP/BOOTP discover packets being sent by the client (this will take thirty to forty seconds). If you do not see any DHCP/BOOTP discover packets, your network (or your network analyzer) is not setup correctly. (How to configure your network this is outside the scope of this document. Figure it out :-)

Now that you have seen packets being sent by the client. Do you see any BOOTP reply packets being sent by the BOOTP daemon? If not, the BOOTP daemon is not properly installed/configured (skip to the next paragraph). If you do see packets being sent by the BOOTP daemon, it sounds like you have a bad cable/connection somewhere or a bad NIC in your client. Replace wires and NICs until things work.

Check that the MAC address(es) in the `/etc/bootptab` file are correctly typed in (only 12 hexadecimal digits, no spaces, dashes, dots or colons). Also, check that the correct MAC address(es) are entered (no typos, transposed digits, etc.).

Enter the following command on your BOOTP server:

```
route
```

This will display the kernel IP routing table. There needs to be an entry that looks like this:

```
255.255.255.255    *        255.255.255.255    UH      0        0        1        eth0
```

If you do not see this line displayed, you probably made a mistake editing the `/etc/rc.d/rc.local` file. If you do not see any mistakes in the file, you probably missed something when installing/configuring Linux. Start over and double check your Linux settings.

The last thing is that the BOOTP daemon is not actually running. Check that `bootpd` was correctly typed in your `/etc/rc.d/rc.local` file. If so, check to see that `bootpd` is running by entering the following command:

```
bootpd -d 3
```

You should see the following information:

```
bootpd: info(6): reading "/etc/bootptab"
```

```
bootpd: info(6): read N entries (N hosts) from "/etc/bootptab"
```

```
bootpd: error(3): bind: Address already in use
```

If you do not get any errors, BOOTP was not running (now it is, try booting your client). If you get any errors besides "Address already in use", there is something wrong in your `/etc/bootptab` file, fix it (it should end up looking like the sample file `/pxe/samples/etc/bootptab`).

2.4.9 Configure NFS

Start the Linux configuration program:

```
linuxconf
```

Select [Quit] to get past the introduction screen.

Select "Config -> Networking -> Server tasks -> Exported file systems (NFS)". Select [Add] to add a new client definition. (The example given here is for the PC1 and PC2 machines from the sample /etc/bootptab given earlier in this document. Don't worry that /nfsroot does not exist, we will create it later.)

```
Path to export:    /nfsroot/pc1
Comment (opt):    (informational only, can be left blank)
Client name(s):   pc1
                  [X] May write
                  [X] Root privileges
                  [ ] translate symbolic links
                  [X] Request access from secure port
```

When you are done, select [Accept]. Select [Add] to add a new client definition.

```
Path to export:    /nfsroot/pc2
Comment (opt):    (informational only, can be left blank)
Client name(s):   pc2
                  [X] May write
                  [X] Root privileges
                  [ ] translate symbolic links
                  [X] Request access from secure port
```

When you are done, select [Accept]. Select [Add] to add a new client definition.

```
Path to export:    /usr
Comment (opt):    (informational only, can be left blank)
Client name(s):   pc1 pc2
                  [X] May write
                  [X] Root privileges
                  [ ] translate symbolic links
                  [X] Request access from secure port
```

When you are done, select [Accept]. Select [Add] to add a new client definition.

```
Path to export:    /home
Comment (opt):    (informational only, can be left blank)
Client name(s):   pc1 pc2
                  [X] May write
                  [ ] Root privileges
                  [ ] translate symbolic links
                  [X] Request access from secure port
```

When you are done, select [Accept].

Select [Quit] to exit from the "Exported file systems" screen.

Select "Config -> Networking -> Misc -> Information about other hosts". You will need to enter the IP address(es) and name(s) of the machines on your network. (It is a good idea to add all of your server and client machines here.) Follow directions to [Add] new definitions.

```
Primary name + domain:  pc1.yourdomain.name
Aliases (opt):          pc1
IP number:              w.x.y.z1
Comment (opt):          (informational only, can be left blank)
```

When you are done, select [Accept]. Select [Add] to add a new definition.

```
Primary name + domain:  pc2.yourdomain.name
Aliases (opt):          pc2
IP number:              w.x.y.z2
Comment (opt):          (informational only, can be left blank)
```

When you are done, select [Accept]. Select [Add] to add a new definition.

```
Primary name + domain:  pxeservername.yourdomain.name
Aliases (opt):          pxeservername
IP number:              w.x.y.z3
Comment (opt):          (informational only, can be left blank)
```

When you are done, select [Accept].

Select [Quit] to exit from the "/etc/hosts" screen.

When you are all done, select [Quit].

Select "Activate the Changes" and press <Enter>. (Nothing happens right away. Just give it a minute.)

You should not get any error messages. If you do, you either made a mistake entering information while in linuxconf or made a mistake installing Linux. Start over and follow the installation instructions.

2.4.10 Edit /etc/hosts.allow

You will need to add your clients to the /etc/hosts.allow file (it is also a good idea to enter the server names for debugging NFS problems). Copy the machine name(s) from the /etc/hosts file to the /etc/hosts.allow file. Only copy the machine names, you do not have to copy the IP addresses or aliases.

```
nfsservername.yourdomain.name
pxeservername.yourdomain.name
pc1.yourdomain.name
pc2.yourdomain.name
```

These names need to match the names in the /etc/hosts file. If you have add names to the /etc/hosts.allow file that are not in the /etc/hosts file, add them to the /etc/hosts file.

2.4.11 End of part 1

Looking good! Time to setup the PXE server.

2.5 PXE Server Setup

Login as root.

2.5.1 Install Kernel Source

The kernel source is not installed in the default RedHat Linux installation. You will need to mount your RedHat CD#1 and install the kernel source packages.

```
mount /dev/cdrom /mnt/cdrom
rpm -Uvh /mnt/cdrom/RedHat/RPMS/kernel*
umount /mnt/cdrom
```

(Ignore the error messages about the kernel packages that are already installed.)

2.5.2 Install & Build PXE & MTFTP Daemons

Build and install the PXE & MTFTP daemons by executing these commands.

```
mkdir /pxe
cd /pxe
mount /dev/fd0 /mnt/floppy
tar xfovz /mnt/floppy/pxesdk.tgz
umount /mnt/floppy
make
make install
```

(Ignore the compiler warnings. Errors will stop the compilation process.)

The PXE and MTFTP daemons have been installed. They now need to be configured.

2.5.2.1 Files installed by PXE "make"

| Filename | Description |
|---------------------------|-------------------------------------|
| /usr/lib/libpxelib.so.1.0 | PXE daemon class library. |
| /usr/lib/libpxelib.so | Linked to /usr/lib/libpxelib.so.1.0 |

2.5.2.2 Files installed by PXE "make install"

| Filename | Description |
|------------------------------------|--|
| /tftpboot/X86PC/UNDI/linux/linux.0 | PXE initial environment layer for Linux remote boot. |
| /usr/sbin/in.mtftpd | MTFTP daemon. |
| /etc/mtftpd.conf | MTFTP daemon configuration file. |
| /usr/lib/libpxeparser.so.1.0 | PXE packet parser class library. |
| /usr/lib/libpxetester.so.1.0 | PXE client tester class library. |
| /usr/lib/libbstrapopts.so.1.0 | Bootstrap option parser class library. |
| /usr/lib/libapitestopts.so.1.0 | API test option parser class library. |
| /usr/lib/libpxeparser.so | Linked to libpxeparser.so.1.0 |
| /usr/lib/libpxetester.so | Linked to libpxetester.so.1.0 |
| /usr/lib/libbstrapopts.so | Linked to libbstrapopts.so.1.0 |
| /usr/lib/libapitestopts.so | Linked to libapitestopts.so.1.0 |
| /usr/sbin/pxe | PXE daemon. |
| /etc/pxe.conf | PXE daemon configuration file. |

2.5.3 Edit /etc/services

Add these lines to the end of the /etc/services file:

```
mtftp      1759/udp
pxe        67/udp
pxe        4011/udp
```

2.5.4 Edit /etc/inetd.conf

Locate the TFTP daemon line (search for "#tftp") in this file and uncomment it. You will also add an entry for the MTFTP daemon at this time.

Change this line:

```
#tftp dgram udp    wait  root  /usr/sbin/tcpd    in.tftpd
```

To these lines:

```
tftp dgram udp    wait  root  /usr/sbin/tcpd    in.tftpd    /tftpboot
mtftp dgram udp    wait  root  /usr/sbin/tcpd    in.mtftpd    /tftpboot
```

2.5.5 Edit /etc/rc.d/init.d/inet

To enable the PXE daemon on restart the /etc/rc.d/init.d/inet file needs to be altered. Look for this line:

```
daemon inetd
```

Add this line immediately after it (don't put "daemon" on the front):

```
pxe
```

2.5.6 Edit /etc/rc.d/rc.local

Add these lines to the end of your /etc/rc.d/rc.local file:

```
route add -host 255.255.255.255 eth0
```

```
route add -net 224.0.0.0 netmask 224.0.0.0 eth0
```

2.5.7 Test Your Changes

Restart your PXE server. When the system is restarted. Login to root and then boot a client machine. The client should start, display its MAC and IP addresses, followed by this network boot prompt:

```
Press <F8> to view menu ...
```

Press <F8> and you will get this network boot menu:

```
Local Boot
```

```
Linux NFS Boot
```

If you do not get the <F8> network boot prompt, the PXE services were not configured correctly or have not started. Check the changes in your /etc/services, /etc/inetd.conf, /etc/rc.d/rc.local and /etc/rc.d/init.d/inet files. If all of these files look good, try rebuilding the PXE services making sure that you do not get any build errors.

```
make remove
```

```
make
```

```
make install
```

Until you get this network boot prompt there is no point in going any further.

Select Linux NFS Boot and press <Enter>. You should see the following go by on the screen.

```
MD w.x.y.z
```

```
BOOT SERVER IP: w.x.y.z
```

```
MTFTP..
```

```
Intel Linux NBP, PXE-2.0 Beta-x (build nnn)
```

```
Downloading linux kernel image
```

```
PXE-T01: File not found.
```

The important message here is the Intel Linux NBP message. If you do not get this message, there is something wrong with the TFTP/MTFTP daemon configuration. Go back and check your changes in the /etc/services, /etc/inetd.conf, /etc/rc.d/rc.local and /etc/rc.d/init.d/inet files. If all of these files look good, try rebuilding the PXE services making sure that you do not get any build errors.

```
make remove
```

```
make
```

```
make install
```

Until you get the Intel Linux NBP message, there is no point in going any further.

2.5.8 Build Client Kernel Image

This section is written assuming that the Linux kernel source is located in the /usr/src/linux directory. If this is not the case, you will need to adjust the commands accordingly. Also, the patches are for RedHat Linux 5.2 (kernel 2.0.36). If you are using a different kernel version or have applied any other patches to the kernel, these patches might not work.

First you need to create this device:

```
mknod /dev/nfsroot b 0 255
```

Next you need to mount your "patch" diskette:

```
mount /dev/fd0 /mnt/floppy
```

Execute these commands, be sure to check for any error messages. Also, the [/pxe/samples/usr/src/linux/make.config](#) file that we include in the PXE Linux SDK is stripped down to the basics. If you want anything exotic (sound cards, multi-port serial devices, printers, etc.), you will need to change your kernel configuration. It is strongly recommended that you start with the kernel configuration included in the PXE Linux SDK and make changes after you get this minimal installation working.

```
cd /usr/src
gzip -cd /mnt/floppy/patch-non640k-2.0.36.gz | patch -p0
umount /mnt/floppy
gzip -cd /pxe/client/patches/patch-pxeundi-2.0.36.gz | patch -p0
cd linux
cp /pxe/samples/usr/src/linux/make.config .config
make config
```

Use the default answers for "make config" (just press <Enter> to all questions), do not make any changes the first time around. **Important note!** If you are remote booting with NICs that are not compatible with the [Preboot eXecution Environment \(PXE\) Version 2.0](#) specification, you will need to include them when you are configuring the kernel. (If your NIC identifies itself with a version number between 0.92 and 0.99n, it is not PXE-2.0 compatible.)

```
make dep ; make clean ; make
make bzImage
cd arch/i386/boot
rdev bzImage /dev/nfsroot
cp bzImage /tftpboot/X86PC/UNDI/linux/linux.1
```

2.5.9 Test Client Kernel

Reset your client again, press <F8> when you see the network boot prompt, select [Linux NFS Boot](#) and press <Enter>. You should see the following messages:

```
Intel Linux NBP, PXE-2.0 Beta-x (build nnn)
Downloading linux kernel image
```

The remote kernel boot should end with some messages like this:

```
Sending BOOTP requests... OK
Root-NFS: Got BOOTP answer from w.x.y.z, my address is w.x.y.z1
Root-NFS: Server returned error 2 while mounting /nfsroot/pcl
VFS: Unable to mount root fs via NFS, trying floppy.
VFS: Cannot open root device 02:00
Kernel panic: VFS: Unable to mount root fs on 02:00
```

This is good because we have not yet created the NFS root file system for the clients. We will do this in the next section. If you do not get this information, something failed during the kernel build. Try again.

2.6 BOOTP/NFS Server Configuration (part 2)

2.6.1 Create /nfsroot

Create the root directory for the client machine you tried to boot. The directory name is displayed in the Root-NFS error message you just got on the client (this is the same directory as the root path that was defined in your /etc/bootptab file). Run the following commands:

```
mkdir -p /nfsroot/pc1
cd /nfsroot/pc1
cp -a /bin /dev /etc /lib /root /sbin /var.
mkdir home proc server tmp usr
cd etc
```

Do not boot your client machine at this time. You need to make some changes to the files in the /nfsroot/pc1/etc directory. You will need to duplicate these instructions for each remote boot client.

2.6.2 Replace /nfsroot/pc1/etc/rc.d/rc.sysinit

You will find a replacement for this file in /pxe/samples/etc/rc.d/rc.sysinit. Use this file only in the client (/nfsroot/pc1/etc/rc.d) directories. Make sure the file is executable.

```
cd rc.d
cp /pxe/samples/etc/rc.d/rc.sysinit rc.sysinit
chmod +x rc.sysinit
```

All of the file system check (fsck) code has been removed from this file and an instruction to mount the NFS partitions has been added.

2.6.3 Replace & Edit /nfsroot/pc1/etc/fstab

You will also need to replace and edit the /nfsroot/pc1/etc/fstab file. You will find it in /pxe/samples/etc/fstab. If you have changed the client machine names, or the name of the nfsroot directory, you will need to make some changes to this file.

```
cd /nfsroot/pc1/etc
cp /pxe/samples/etc/fstab fstab
```

Below is a sample etc/fstab file.

| | | | | |
|------------------------|-------|------|----------|-----|
| nfsserver:/nfsroot/pc1 | / | nfs | defaults | 1 1 |
| nfsserver:/usr | /usr | nfs | defaults | 1 1 |
| nfsserver:/home | /home | nfs | defaults | 1 1 |
| none | /proc | proc | defaults | 0 0 |

2.6.4 Remove /nfsroot/pc1/etc/mtab

```
rm /nfsroot/pc1/etc/mtab
```

2.6.5 Edit /nfsroot/pc1/etc/HOSTNAME

Change the name in this file to match the name of your client machine.

2.6.6 Edit /nfsroot/pc1/etc/sysconfig/network

Change the HOSTNAME line in this file so it also matches your client machine name.

2.6.7 Boot A Client

Boot the client machine and login as root. You should see this:

```
pc1 login: root
```

Last login: Fri Apr 30 13:34:59 on tty1

[root@pc1 /root]#

See the next section for troubleshooting if you are unable to connect back to the NFS server and login.

2.6.8 NFS + PXE Troubleshooting

Since we have tested everything up to the final mounting of the NFS shares and the changes made in the /nfsroot/pc1/etc directory, the problem probably lies here. Before you go hunting, try rebooting both servers and the client one more time, and watch the network traffic on the analyzer. Verify that the client is booting from the correct servers and downloading the expected files.

If the network traffic looks correct, go back and double check the /nfsroot/pc1/etc/fstab, /nfsroot/pc1/etc/HOSTNAME, /nfsroot/pc1/etc/sysconfig/network or /nfsroot/pc1/etc/rc.d/rc.sysinit files. If all of these are correct, a mistake was made earlier that did not show up until now. (Probably a selection was made, or missed, when installing Linux or configuring the kernel. Do over ;-)

3. Configuration of PXE and MTFTP Daemons

3.1 Configuration File for PXE Daemon

All of the PXE specific configuration information is stored in the /etc/pxe.conf file on the PXE server. The following table shows the major file entries and their default values. All other file entries, and their descriptions can be found in the /etc/pxe.conf file.

| Entry Name | Default Value | Description |
|--------------------------|---------------|---|
| Network_Interface_Name | eth0 | Name of the network interface. |
| Mtftpd_Root_Directory | /tftpboot | Base directory for the MTFTP files. Use the same base directory for the TFTP files so the fall-back mechanism in the PXE boot ROMs will work. |
| UseDHCPPort | 1 | Set this to 1 if there is no DHCP or BOOTP server running on this system. If there is a DHCP or BOOTP server running on this system this must be set to 0, and the BOOTP/DHCP must return a class identifier (option 60) of "PXECient". |
| StartProxy | 1 | Start the PXE Proxy service in the PXE daemon. This enables the PXE daemon to respond to PXE Proxy discover and request packets. |
| StartBootService | 1 | Start the PXE Boot service in the PXE daemon. This enables the PXE daemon to respond to Boot service discoveries, if it has the requested images. |
| Discovery_BCast_Disabled | 0 | Set to 1 to disable Boot service broadcast discovery. This must be set to 1 if UseDHCPPort is set to 0. |
| Discovery_MCast_Disabled | 0 | Set to 1 to disable Boot service multicast discovery. |
| Discovery_MCast_Addr | 224.0.1.2 | Boot service multicast listening address. This should be the same on all boot servers. |

3.2 Configuration File for MTFTP Daemon

All of the MTFTP specific configuration information is stored in the /etc/mtftpd.conf file on the PXE server. The following table shows the major file entries and their default values. All other file entries, and their descriptions can be found in the /etc/mtftpd.conf file.

| Entry Name | Default Value | Description |
|------------------------|---------------|--|
| IsMulticastEnabled | 1 | If set to 1, multicast support is enabled. If set to 0, normal TFTP operation ensues. |
| ServerPort | 1759 | UDP port on the MTFTP server to which the clients must send the initial MTFTP open request packet. |
| ClientPort | 1758 | UDP port on the client to which the server will send the MTFTP responses. |
| IP_Multicast_TTL | 32 | TTL value in the IP header for MTFTP packets. |
| Multicast_ip_addresses | varies | Multicast IP address assignments for all files. |

4. Revision History

| <i>PXE SDK Release Date</i> | <i>PXE SDK Ver #</i> | <i>Wired for Management Baseline Version</i> |
|-----------------------------|----------------------|--|
| 5/4/99 | V3.0b1 bld 001 | V2.0 |
| Documentation | | |
| • Initial release. | | |
| PXE Daemon | | |
| • Initial release. | | |
| MTFTP Daemon | | |
| • Initial release. | | |
| Linux.0 NBP | | |
| • Initial release. | | |